

# Artificial Intelligence

## in the Oilfield

### A Schlumberger Perspective

**Reid G. Smith**

**Schlumberger Laboratory for Computer Science**

**Austin, Texas**



Laboratory for Computer Science

Keynote presentation. *Artificial Intelligence in Petroleum Exploration and Production Conference*, Texas A&M University, College Station, TX, USA, May, 1990.

Several graphical slides are not included.

# Artificial Intelligence

## Engineering View

**Development of computational architectures that use knowledge separate from algorithms in problem solving.**

 Schlumberger Laboratory for Computer Science

First of all, let's remind ourselves of what we are talking about when we use the term "Artificial Intelligence."

This definition is one given me by Raj Reddy at Carnegie-Mellon University.

Certainly, there are other ways to define AI. For example,

as the attempt to construct artifacts that exhibit behavior we call "intelligent behavior" when we observe it in human beings,

or, as the development of a systematic theory of intellectual processes, wherever they are found.

However, I prefer what I have called the "Engineering View" on the slide. It focuses our attention on what the "suppliers" and "appliers" of AI technology actually do.

## AI ...circa 1975

### Application Areas

#### Core Topics

**Heuristic Search**

**Knowledge Representation**

**Commonsense Reasoning & Problem Solving**

**Architectures & Languages**

**Game Playing**

**Theorem Proving**

**Robotics**

**Natural Language**

**Expert Systems**

**Automatic Programming**

**Vision**

**Information Processing Psychology**

**Schlumberger**

Laboratory for Computer Science

To begin our tour, let's situate ourselves in the AI field as it was seen around the mid-seventies.

This breakdown into core topics and application areas is taken from a survey by Nils Nilsson, then at SRI, now Chairman of the Computer Science Department at Stanford University.

At the time, a relatively new area called "Expert Systems" had emerged, along with some rudimentary but useful technology. This technology had been derived to a large extent from experiments done at Stanford in the areas of Chemistry and Medicine—with results embodied the Dendral and Mycin programs respectively.

It was in this context that Schlumberger became interested in AI. In 1978 a conference on "Knowledge Engineering applied to Schlumberger Interpretation" was held at our research laboratory in Connecticut.

This conference brought together several AI luminaries—Ed Feigenbaum, Randy Davis, Peter Hart, Pat Winston, Raj Reddy, Bruce Buchanan, Penny Nii...—with Schlumberger interpretation specialists—Jean Dumanoir, George Coates, Chris Clavier, Al Gilreath...

The result was selection of a first application of AI techniques—interpretation of data from the dipmeter tool.

# Dipmeter Advisor

## Input

well logs

geological assertions

## Output

structural dip

tectonic & stratigraphic feature analysis



Laboratory for Computer Science

The resulting system was called the Dipmeter Advisor.

# **Dipmeter Advisor ...Accomplishments**

- **Consistent interpretations within its expertise**
- **Vehicle for codification of expertise**
- **Provocation of discussion among specialists**
- **Interpretation laboratory**
- **Interactive workbench for manual interpretation**

 Schlumberger Laboratory for Computer Science

# Code Distribution

<b>Inference Engine</b>	<b>8 %</b>
<b>Knowledge Base</b>	<b>22 %</b>
<b>Feature Detection</b>	<b>13 %</b>
<b>User Interface</b>	<b>42 %</b>
<b>Support Environment</b>	<b>15 %</b>

This is the slide that changed my direction for a few years!

# Lessons Learned

- **Performance & Integration**
- **Technology Transfer**
- **Intelligent Assistant Metaphor**

 Schlumberger Laboratory for Computer Science

In the next few minutes I will try to relate to you some lessons we learned from the Dipmeter Advisor System.

There are basically three lessons, shown on this slide.

# Lessons Learned

- **Performance & Integration**

numeric & symbolic components

standard platforms

standard programming languages



Laboratory for Computer Science

# Lessons Learned

- **Technology Transfer**

standard platforms

standard programming languages

software engineering

...documentation

...training



Laboratory for Computer Science

# The Overall Problem

**Graphics**  
**User Interface**  
**Numerical Processing**  
**File & Data Management**  
**Communications**  
...

**AI**

 Schlumberger

Laboratory for Computer Science

This slide shows why we shouldn't be surprised by the first two lessons. After all, we were actually only trying to solve a rather small part of the overall problem. Even if the knowledge base and inference engine had performed perfectly, we still would have been left with graphics, user interface, ...

After the Dipmeter Advisor System, one thing we set out to do was attack some of the other software components on this slide.

**"The greatest benefits to be derived from a computer will probably consist of information impossible to obtain previously ... "**

**"Our experience has shown that the computer is more adaptable to some projects than others ... "**

**"It is impossible to overemphasize the desirability of providing for convenient corrections or deletion of errors in data ... "**

**"The maximum justifiable amount of flexibility for extending or integrating applications must be included in the initial programming ... "**

***G.M. Sheehan, Proc. Automatic Data Processing Conf., Sept. 1955.***

 Schlumberger Laboratory for Computer Science

This slide just to show that AI types were not the first to see problems in computerization...note the date, 1955!

# Lessons Learned

- **Performance & Integration**
- **Technology Transfer**
- **Intelligent Assistant Metaphor**

 Schlumberger Laboratory for Computer Science

Note that the first two lessons are not limited to AI...one could make similar comments about many software technologies.

The third lesson, however, was directed to AI.

# Lessons Learned

- **Intelligent Assistant Metaphor**

## Domain-Specific User Interaction

... end-user customization

... knowledge acquisition

We were struck by the importance of user interaction, both because of the enormous variety of graphical displays asked for by the interpreters, but also because of the importance of user interfaces for knowledge acquisition.

# Traditional Architectural View

Application

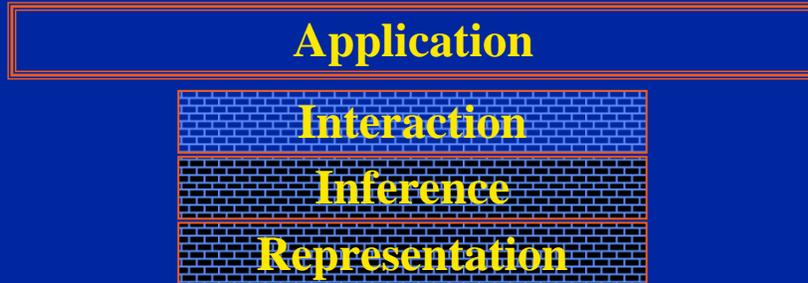
Inference

Representation

Schlumberger

Laboratory for Computer Science

# Revised Architectural View



 Schlumberger Laboratory for Computer Science

One practical reason for making this argument is that many of the problems one sees in petroleum exploration are not going to submit to fully automatic solutions any time soon. Consequently, we will have humans in the loop. And if we are going to have humans in the loop, we had better be very concerned with how our systems interact with them.

# AI Technology ⇨ User Interface

- **Object-Oriented Interface Constructs**
- **Interpretation of Knowledge Base to Specialize Views & Interaction Methods**
- **Constraints to Maintain Consistency**
- **Inference of Missing or Dependent Information**

 Schlumberger Laboratory for Computer Science

Luckily we had some technology at hand to help solve the problem. In fact, much of the technology that we were already using to encode the domain knowledge and use it.

# User Interface ⇨ Knowledge Acquisition

- **Expression & Interaction in Domain Terms**
- **Focus on Domain Knowledge & Problem-Solving Methodology**
- **Explanation & Debugging**



Laboratory for Computer Science

## AI ...circa 1975

### Application Areas

#### Core Topics

**Heuristic Search**

**Knowledge Representation**

**Commonsense Reasoning & Problem Solving**

**Architectures & Languages**

**Game Playing**

**Theorem Proving**

**Robotics**

**Natural Language**

**Expert Systems**

**Automatic Programming**

**Vision**

**Information Processing Psychology**

**Schlumberger**

Laboratory for Computer Science

## AI ...1990

### Automated Reasoning

automatic programming, planning & scheduling, rule-based reasoning, search, theorem-proving, uncertainty, truth-maintenance & constraint-based systems

### Commonsense Reasoning

qualitative reasoning, design, diagnosis, simulation

### Knowledge Representation

inheritance, non-monotonic & non-standard logics, temporal reasoning

### User Interfaces

### Knowledge Acquisition & Expert System Design Methodologies

### Architectures & Languages

### Natural Language

### Machine Learning

### Robotics

### Cognitive Modeling

### Education

### Vision

### Neural Networks



Laboratory for Computer Science

Now let's move forward to 1990. This list is taken from the call for papers for the National Conference of the AAAI for this year.

# Knowledge-Intensive Development Environments

## HyperClass®

**Representation:** Object-Oriented Language

**Inference:** Rules, Constraints, ...

**Interaction:** User Interface Development Substrate

Customized Editors

⇒ *Knowledge-Acquisition Substrate*



Laboratory for Computer Science

Based on the experience we had, we became interested in what we called "knowledge-intensive" development environments—for building knowledge-based systems.

The representation and inference substrates contain problem-solving methods, knowledge of the use of individual subsystems, and basic knowledge of the domains in which they are applied. The interaction substrate provides tools for constructing user interfaces. It supports the perspectives of three different types of user: system developer/maintainer, domain specialist, and end user.

Our first attempt at such an environment is called HyperClass. It includes ...

The following sequence of slides shows a number of different views of the geological and interpretation knowledge embodied in a contemporary system. (The system is "virtual" one—I have put together bits and pieces of several systems.)

You will see that each view is aimed at a different type of user.

(I want to emphasize that each of the different views of the domain knowledge is provided by the same user interaction substrate—via a set of customized editors. The key idea that enables us to do this is separation of the domain knowledge from the mechanisms for presenting and interacting with it.)

# HyperClass Slide Sequence

**Normal Fault Object Editor**

**Sedimentary Analysis Rule**

**Tectonic Feature Taxonomic Hierarchy**

**Tectonic Structure Editor**

**Log Graphics Editor**

**Elan Processing Chain**

 Schlumberger Laboratory for Computer Science

**Normal Fault Object Editor:** The first editor shows the concept of a normal fault in a way suited mainly to the developer/maintainer. Parts, attributes, methods, and rules.

(In many of the slides the fixed and pop-up command menus are not shown.)

**Sedimentary Analysis Rule Editor:** Here is another view of the geological knowledge suitable for the developer or the domain specialist. As a side point, note that this is a rule about geology. Unlike the original Dipmeter Advisor System rules, this one does not freely mix geological knowledge with interpretation knowledge—i.e., the way the geology manifests itself in the logging data.

**Tectonic Feature Taxonomic Hierarchy:** Here is another view of the knowledge—suitable for a structural geologist.

**Tectonic Structure Editor:** This editor depicts another way a geologist might look at the knowledge: It is based on work done by John Suppe as Princeton.

This is a "forward modeling" system. We can run the geological reverse faulting model both forward and backward in time. We can then penetrate the structure with boreholes and look at the resulting, simulated data. We will later return to forward modeling.

**Log Graphics Editor:** Here is a traditional way to view the data—from the perspective of the log analyst.

(Note again that each of these different views of the domain knowledge is provided by the same user interaction substrate—via a set of customized editors.)

**Elan Chain Editor:** Another view suitable for a log analyst. Historical view of the interpretation of the data.

# Pleiades

## Object-Oriented Architecture

**integrating:**

**family of applications**

**AI and conventional techniques**

**interactive and batch modules**

**using explicit process representations**

 Schlumberger Laboratory for Computer Science

We now momentarily leave the wellsite with our data and move to the client office. The next problem we set off to deal with was the management and interpretation of all of the log data.

This was a continuation of our attempt to move out of the small "AI" box I showed you earlier into the larger box that is the "rest of the problem" (the graphics, file and data management, etc.).

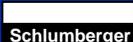
An interesting part of Pleiades is the use of what I have called here "explicit process representations." In an object-oriented fashion, we modeled the task-subtask hierarchy of various Schlumberger processing chains. Indeed the Processing History editor I showed you a minute ago was based on this process representation. These models also include descriptions of the inputs & outputs of the various processing modules. This enabled us to separate the processing algorithms from the interaction mechanisms. Previously these had been freely mixed in our interpretation code—with a resulting loss of consistency and reusability.

**Next Slide:** Perhaps the main thing we see on this slide is that during the early eighties the color workstation was invented!

**Next Slide:** On this screen we see that the scrolling log graphics presentations originally developed for the Dipmeter Advisor System had caught on for many types of data...here, the casing and the tools that had been run into the hole.

# Log Quality Monitoring System

- **Real-time anomaly detection and resolution**
- **Reduction of engineer information overload**
- **Explicit models of tool behavior and borehole environment**
- **Knowledge acquisition by domain experts (via OSR nets)**

 Schlumberger Laboratory for Computer Science

Preamble Slide: The field engineer sitting at the display is asking his colleague... "What went wrong in this interval?"... or words to that effect.

Our next system is called "LQMS—The Log Quality Monitoring System." No doubt many of you have heard Dennis O'Neill or other members of the development team discuss it, either at the last year's conference, or at other conferences.

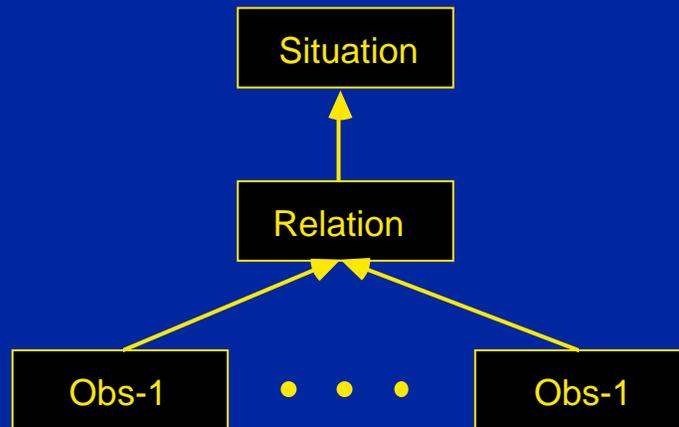
Suffice it to say that Schlumberger is very interested in providing high quality data to its clients... and we try hard to avoid situations like the one I caricature here. To aid in the process we are experimenting with a system to detect and resolve anomalies in the data while we are logging a well.

The goals are:

To provide the field engineer with the assistance necessary to decide whether a log needs to be re-run before leaving the wellsite, and

To provide the client with the information necessary to successfully interpret the data in the absence of outright tool failures

# OSR Nets



- **Relation(Obs1,...,Obsn)  $\Rightarrow$  Situation.**

 Laboratory for Computer Science

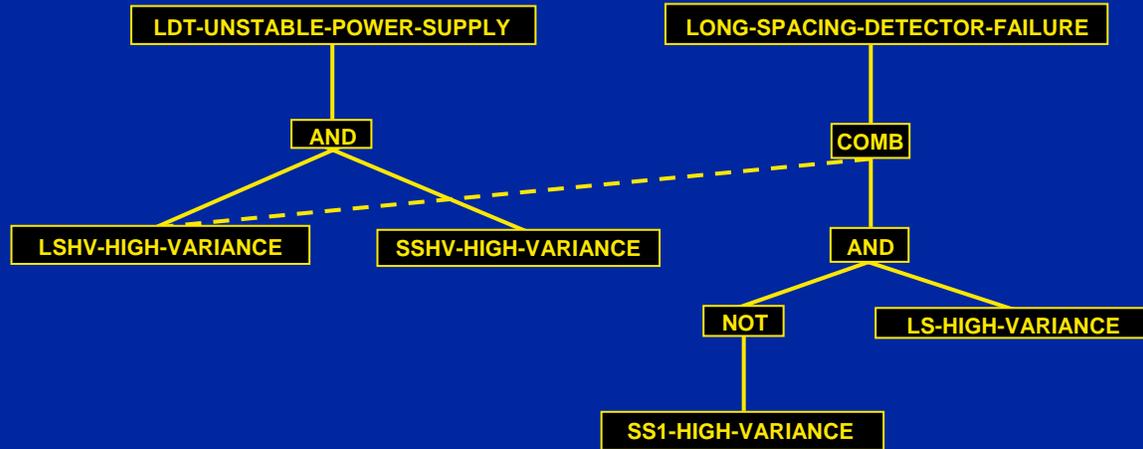
One of the key developments during this project has been what we call OSR Nets—for Observation, Situation, Relation. The idea is that combinations of observations made on the data suggest the presence of situations—the underlying causes of the observations. The relations are simple logical connectives, OR, AND, NOT, and COMB (a weighted average).

OSR nets are a specialized inference mechanism...tuned to the way we found that our field engineers viewed the problem. In fact, OSR nets were suggested by the field engineers themselves.

The network presentation shown here is essentially the editing interface given to the field engineers. It is what they have thus far used themselves to construct the LQMS knowledge base—some 2,000 classes that encode qualitative models of over a dozen tools and their associated borehole effects.

Once again we have seen the importance of a user interface that allows the domain specialist to add, browse, and modify the knowledge in "domain" terms. One might observe that this has been a trend throughout the application of AI techniques. For example, Rules and objects (or "frames" as they were then called) were argued for in part because they let the domain specialist get a little closer to the encoded knowledge than did Lisp or C code, thus reducing (slightly) the knowledge acquisition bottleneck.

# OSR Nets



Schlumberger Laboratory for Computer Science

Here is an example network for the Litho-Density Tool.

LSHV-HIGH-VARIANCE (variance in the long-spacing high voltage channel) is a symptom indicative of one of two problems, either an unstable power supply or a long-spacing detector failure. (The LDT has two sensors, called long-spacing and short-spacing—that share a power supply.)

If both high voltage channels have high variance, the problem is likely to be an unstable power supply.

On the other hand, if there is a combination of evidence, including variance in the long-spacing high voltage channel and variance in the long-spacing measure channel, but not including variance in the short-spacing measure channel, then the problem is likely to be a long-spacing detector failure.

This information first tells the field engineer that there is a problem with the data. Second, it tells him which module to have shipped to the wellsite. In either case, it helps prevent driving away from the wellsite—only to find later that the data is problematic.

# Slurry Design Assistant

- **Supports a consistent, worldwide cement slurry design methodology**
- **Provides an intelligent design module for CemCADE™ to assist in the design of optimal cement slurries**
- **Combines heuristic, object-oriented, and procedural elements**

 Schlumberger Laboratory for Computer Science

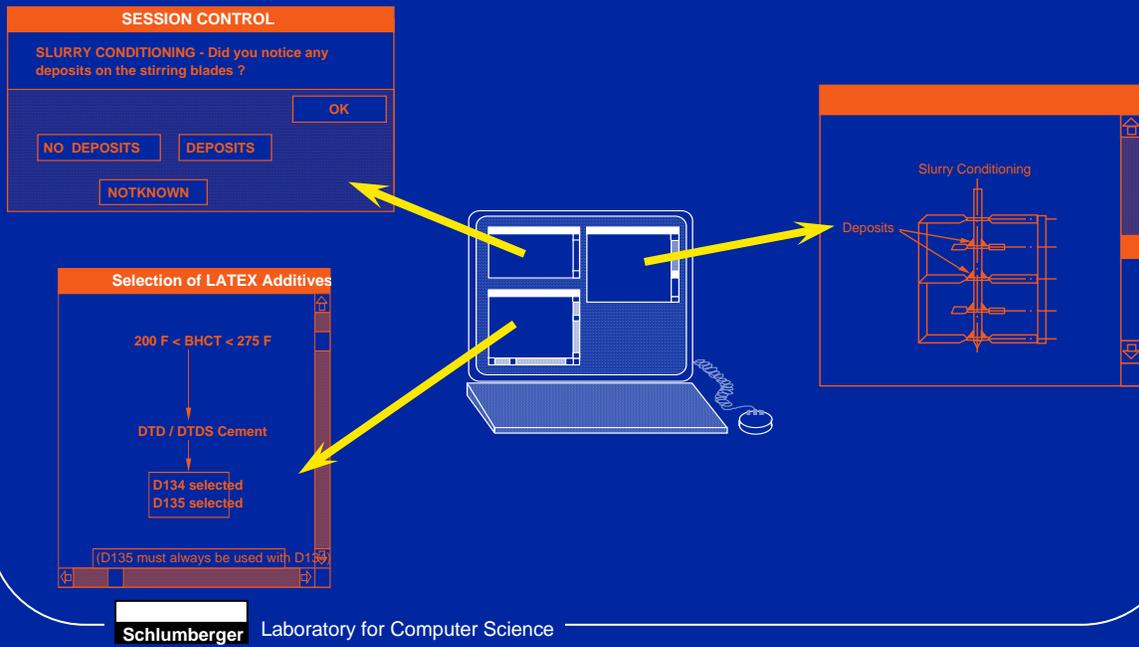
Having successfully interpreted all of the data, we now move on to cementing the casing.

A cement slurry consists of the cement powder, some type of mix water (i.e., fresh, sea, or salt water), and one or more chemical additives that give the cement specified physical properties—like time required to harden, compressive strength, etc.

Central to this is choosing the appropriate additives for the cement slurry.

To this end, we are building a Slurry Design Assistant.

# Slurry Design Assistant



Here we see the system asking some questions about the current situation. "Did you notice any deposits on the stirring blades?"

It uses this information to help determine which type of latex additive to recommend. The small window on the right helps the user visualize the situation referred to in the question.

The window on the bottom shows a portion of the reasoning chain.

# Slurry Design Assistant

## Implementation technology:

- **PC and workstation-based platforms**
- **NEXPERT *OBJECT* for rules and object representations**
- **Relational database management system**
- **FORTRAN for numerical calculations**

 Schlumberger Laboratory for Computer Science

From an implementation point of view, the system looks much like the other systems we have seen, although the relational database is a new twist.

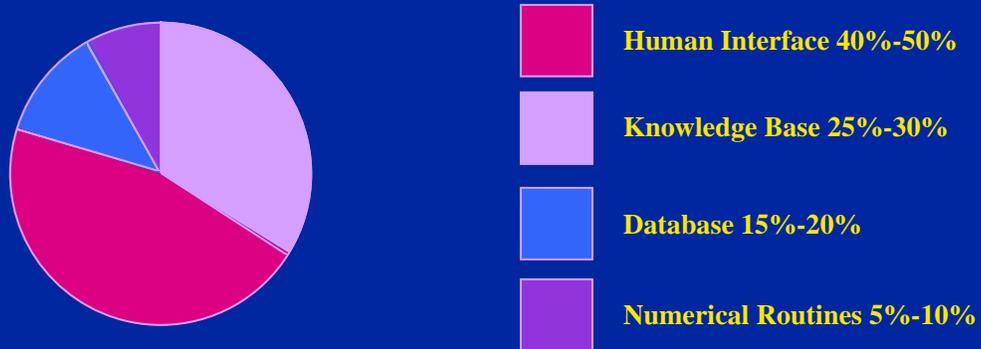
To date we have implemented:

a ruleset for design of gas migration prevention slurries (>400 rules);

an advice mode for improving slurry performance based on lab tests.

The system is currently in field test in Gulf Coast and Europe.

# Slurry Design Assistant



 Schlumberger Laboratory for Computer Science

Here is a breakdown of the code in the Slurry Design Assistant. We see a story that is quite consistent with that of the Dipmeter Advisor System.

# MDS

- **Drilling Process Monitor**
- **Evaluation & Planning Workstation**
  - Drilling Data Analysis, Performance Comparisons,
  - Kick Control, BHA Design, Inventory Control

## Object-Oriented Models & Interfaces

 Schlumberger Laboratory for Computer Science

We now move on to the drilling rig floor, with a system developed by Sedco-Forex called MDS.

There are two parts to the system. The first gives the driller a graphical presentation of the data he needs...including various alarms.

We will discuss the second portion of the system. It is for use by the rig superintendent and oil company representative ...

**Drilling Data Analysis:** The system does lithology evaluation based on rate of penetration.

**Performance Analysis:** It allows trip planning analysis (e.g., hookload by depth in the past 3 trips).

It can generate a variety of reports, customized for the client.

**Drilling Engineering Calculation:** The system has a hydraulics planner...it can deal with pressure losses, cuttings transport, hole cleaning and optimization.

It helps the driller keep track of his equipment.

### Screen Slide:

MDS makes use of the separation of domain knowledge and the ways in which it may be presented to different user groups (e.g., end users, domain specialists, and system developers).

# Two Approaches

*narrow, specialized systems*

**KBSs**

*integrating frameworks*

**System Architectures**

**Object-Oriented Substrate**



Laboratory for Computer Science

**CSI Slide:** Let's step back a bit and characterize the systems we have constructed for Oilfield use.

**Next Slide:** The systems I have described fall into two basic categories:

On one hand, we have narrow, specialized systems like the Dipmeter Advisor System and the Log Quality Monitoring System.

On the other hand, we have integrating frameworks, like Pleiades and MDS.

In all cases, we have found an object-oriented substrate to be a useful building block.

# AI at Schlumberger

**Automated Reasoning**

**Automatic Programming**

**Knowledge Representation**

**User Interfaces**

**Knowledge Acquisition & Expert System Design Methodologies**

**Architectures & Languages**

**Machine Learning**

**Perception & Signal Understanding**

**Vision**

**Expert Systems – Applications**

**Data Interpretation, Diagnosis, Design, Process Planning,  
Materials Selection, ...**



Laboratory for Computer Science

Let's quickly step back even further and look at the AI areas in which we have worked in Schlumberger as a whole—for the moment not restricting ourselves to the Oilfield.

Once again, I have used the call for papers from the 1990 National Conference of the American Association for Artificial Intelligence as a template.

It is apparent that we have covered a wide variety of areas in our laboratories. In terms of application, the work has been aimed at CAD/CAM, ATE, and Electricity Management, as well as Oilfield Services.

As of a year ago, we centralized our research efforts in Computer Science—in Austin. Whereas in the past, we have tended to have groups with "AI" in the name (e.g., Fairchild Laboratory for Artificial Intelligence Research), today we have no such groups. We are focused more on "application areas" (e.g., diagnostic systems for ATE VLSI testers). Each of the groups draws from AI technology as appropriate—and extends the state of the art as necessary. In other words, AI technology has diffused throughout our computer science research efforts—but in close combination with applications and traditional software.



Now let's look into the future. During the remainder of the talk I intend to show you some examples of what we are working on now.

I have three purposes here:

First, I hope to remind you that AI is something more than Expert Systems and Neural Networks. It is clear that these areas have been the major foci of applications efforts and business ventures—**but wait, there's more!**

My second purpose is to try to interest you in experimenting in the areas I will touch upon.

And finally, truth be known, I find these areas interesting, challenging, and fun!

## AI ...1990

### Automated Reasoning

automatic programming, planning & scheduling, rule-based reasoning, search, theorem-proving, uncertainty, truth-maintenance & constraint-based systems

### Commonsense Reasoning

qualitative reasoning, design, diagnosis, simulation

### Knowledge Representation

inheritance, non-monotonic & non-standard logics, temporal reasoning

### User Interfaces

### Knowledge Acquisition & Expert System Design Methodologies

### Architectures & Languages

### Natural Language

### Machine Learning

### Robotics

### Cognitive Modeling

### Education

### Vision

### Neural Networks

Schlumberger

Laboratory for Computer Science

Returning to the NCAI call for papers, we notice that vision is a topic on the list. Of course it was on the 1975 list as well.

From the outset, the attempt has been to provide automatic image interpretation equivalent to human vision. At Schlumberger we have pursued work in two major areas:

First, low-level stereo vision. The result this year is a new image registration system for one of our VLSI test systems.

Second, we have worked on what is now called physically-based modeling. The essential point here is a recognition that computer vision and computer graphics are duals. Vision is the analysis of images and graphics is the synthesis of images.

Physically-based modeling combines the two. We analyze images to determine structure and constraints and based on physical principles (for example, viscosity and plasticity), synthesize a 3D model. This work is today finding application in CAD/CAM and seismic interpretation...but I get ahead of myself.

Why vision in the oilfield?

**Next Slide:** First of all, many of the new logging tools are imaging tools. This slide shows many of them.

# Machine Vision

**Attempt to provide automatic image interpretation equivalent to human vision**

- **Vision, Modeling Duality**
- **Low-Level Vision**

 Laboratory for Computer Science

The next few slides show some of the images the tools produce.

**AIT/FMS Images:** The array induction imager (AIT) makes multiple lateral, high-resolution resistivity profiles of the formation around the well. The resulting radial saturation log shows resistivities up to 60" into the formation in color to highlight pay zones. Saturation is displayed in %—from 35% in red to 100% in blue. This images also shows invasion of the mud filtrate (the black line).

The Formation Microscanner (FMS) measures microresistivity via planar arrays on four pads. It shows core-like detail.

The combination of the two images links petrophysical and geological information.

In order to enhance reservoir characterization, we are also working on integrated interpretation of data at multiple scales—from the km scale of seismic to the cm scale of microresistivity measurements. Once again, images are central to this activity.

**Cement Map:** The ultrasonic imager (USIT) enhances evaluation of cement quality and corrosion. It uses a rotating ultrasonic transducer and sends full waveforms to the surface. The software can assess cementation, monitor ellipticity and degradation of the casing, estimate casing thickness—to monitor corrosion on the outside of the casing, and identify pits and defects in the casing.

Blue is fluid behind the casing, red is gas, and brown is cement.

**Pleiades:** Here we return to the Pleiades workstation. We see some of the displays that can be generated for examining data from the Formation Microscanner.

# Seismic Horizon Extraction

Marked Paper Seismic Sections ⇨ Workstation Database

- **RGB → H-S(-V)**
- **Domain Knowledge**

**Rule-Guided Filtering**



Laboratory for Computer Science

Now let's consider a problem in seismic interpretation. Hand-interpreted seismic sections are widespread...the interpretations are done with colored pencil (one of two essential tools for geologists [the other being the highway systems...that produced such useful outcrops around the world]). The problem is to capture the interpretations in electronic form for inclusion in a workstation database. (Actually this is a problem much more widespread than seismic sections. In many applications, we have a desire to convert the old "paper" system to an electronic system, but the problem of data entry is massive.)

We have only scratched the surface on this application. We start with quite standard image processing techniques. We convert from the Red-Green-Blue encoding from a standard TV camera to the Hue-Saturation(-Value) encoding and then classify according to position in the resulting space. We then "filter" the results using some domain knowledge encoded as a series of rules—rules that know something about seismic horizons.

As a side point, this is an idea that we found quite useful in the Dipmeter Advisor System...using rules to filter and merge basic dipmeter patterns that had been detected with standard signal processing techniques.

## AI ...1990

### Automated Reasoning

automatic programming, planning & scheduling, rule-based reasoning, search, theorem-proving, uncertainty, truth-maintenance & constraint-based systems

### Commonsense Reasoning

qualitative reasoning, design, diagnosis, simulation

### Knowledge Representation

inheritance, non-monotonic & non-standard logics, temporal reasoning

### User Interfaces

### Knowledge Acquisition & Expert System Design Methodologies

### Architectures & Languages

### Natural Language

### Cognitive Modeling

### Machine Learning

### Education

### Robotics

### Vision

### Neural Networks



Laboratory for Computer Science

We have worked in automatic programming for several years, usually associated with writing software to drive our downhole tools. What I'd like to tell you about now is a new effort...aimed at generating forward modeling code.

# Automated Synthesis of Mathematical Modeling Codes

Sinapse

Mathematics → Fortran or C

Extensions to *Mathematica*

 Schlumberger Laboratory for Computer Science

The system being constructed in our laboratory is called Sinapse. The basic notion is simple: try to go as directly as possible from the Mathematics to a standard programming language, such as Fortran or C.

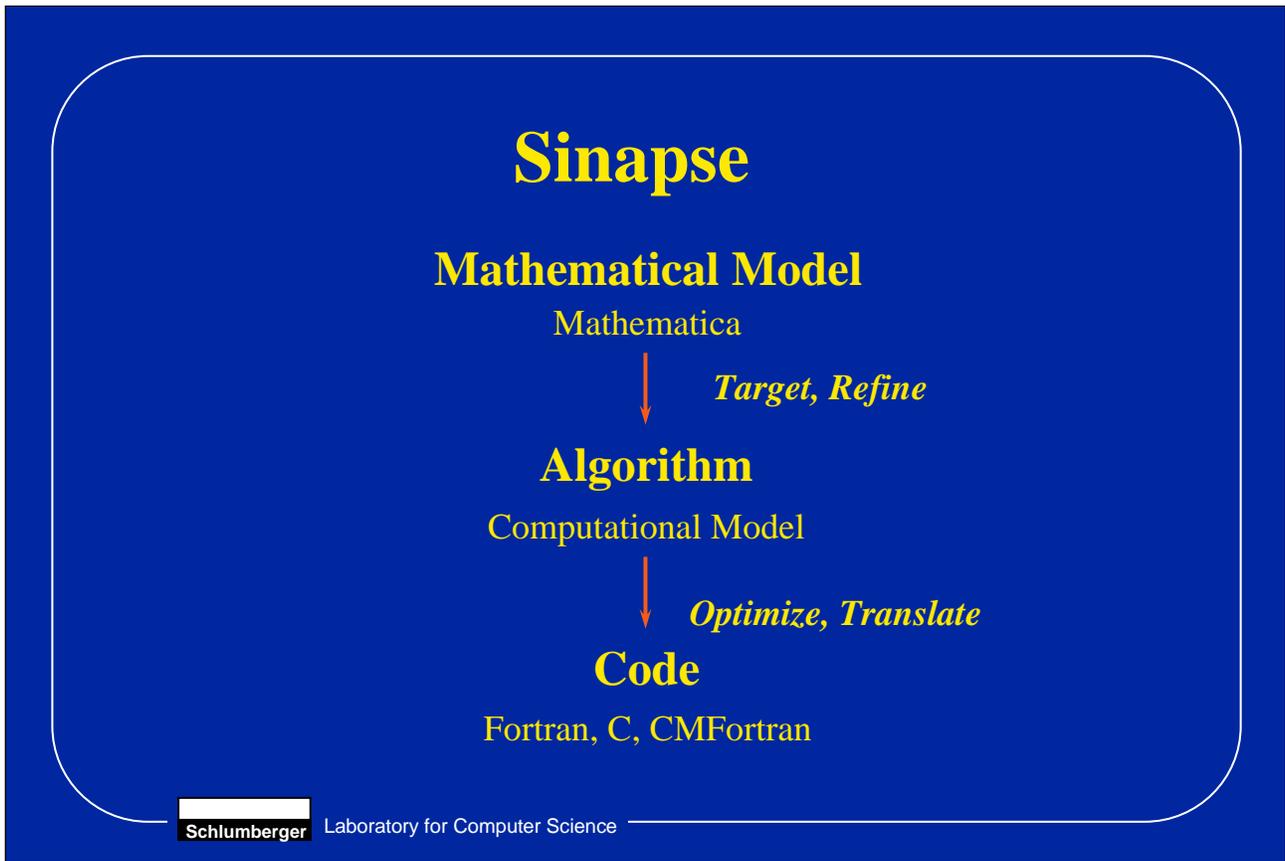
We are interested in this because mathematical modeling is central to Schlumberger's businesses—both in Oilfield Services and in our metering business.

We hope to improve quality by avoiding clerical errors—for example, 3D boundary cases are hard to deal with.

We hope to improve productivity—by permitting quick experimentation with many models—optimizing for stability, accuracy, or efficiency.

We hope to further improve efficiency by optimizing for multiple machines/languages—for example, serial, vector, or parallel machines, and for Fortran, ConnectionMachine-Fortran, or C.

Finally, we seek to improve reusability by having clearer specifications—written in terms of mathematics and physics.



Here is a diagram of the synthesis process.

We start with a model, encoded in equations in Mathematica notation. We select a target architecture and use a refinement system to generate a computational model, encoded in a "generic" programming language. We then optimize and translate into a standard programming language.

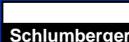
It is perhaps surprising to see how early in the game we start to worry about the target architecture...but upon reflection, it is clear that even the computational model, let alone the code, is quite different for a fine-grained massively parallel machine like a Connection Machine than it is for a vector machine like a Cray.

# Sinapse

## Example: Wave Propagation

- **Elastic and acoustic properties**
- **2D and 3D explicit finite differencing**
- **Many boundary condition alternatives**
- **>10 variants in Fortran77 & Fortran8X**

Specification size	40 - 70 lines
Generated code	200 - 1600 lines
Synthesis time	20 - 120 Sun 4 secs
Run time (FDMME)	.9 times original code

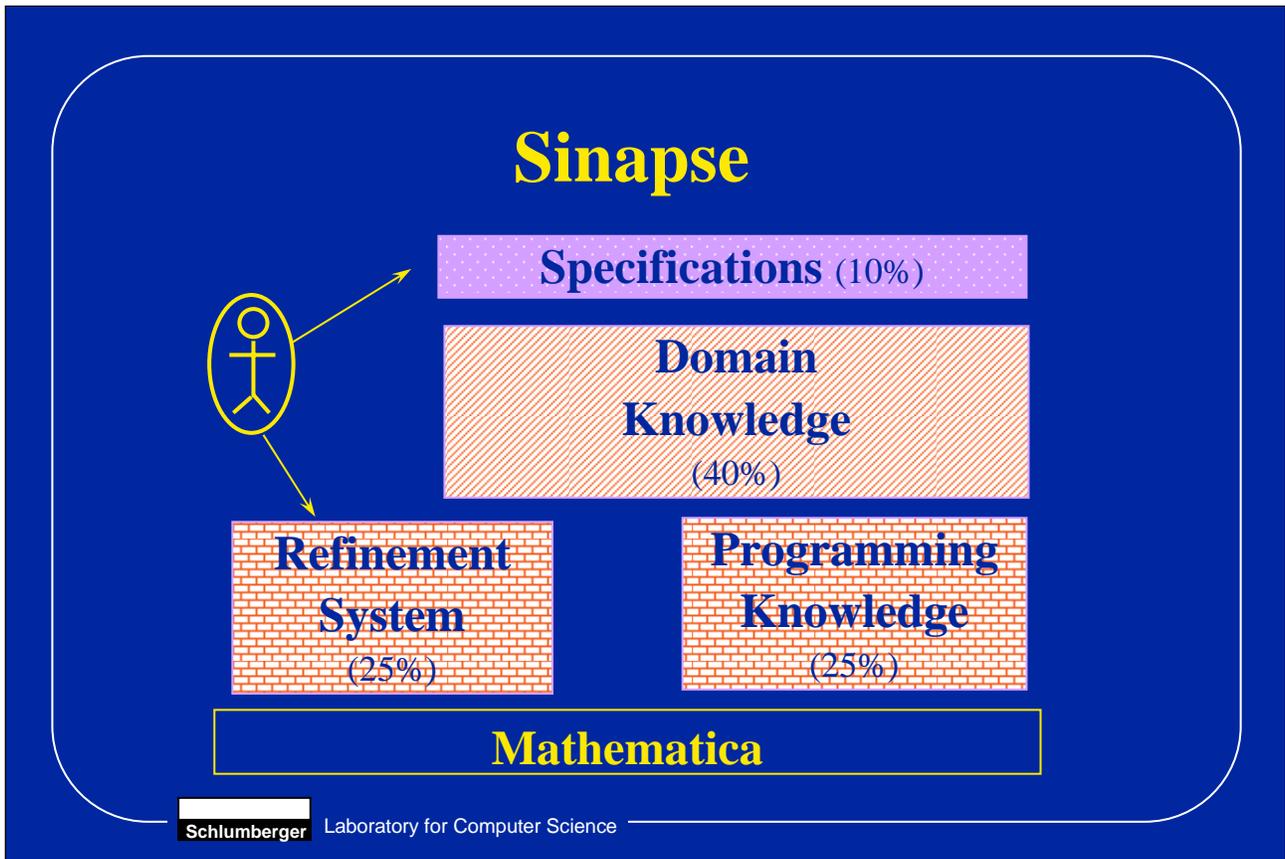
 Schlumberger Laboratory for Computer Science

This slide is a summary of the initial work we did last year. This work was done to explore the basic ideas. Basically, we reverse engineered an existing piece of Schlumberger commercial modeling code—the original code was for 2D elastic wave propagation.

We generated several variants of this code as shown on the slide.

The following sequence of slides shows a result of running the code. You will see a BH2 (Blackman-Harris 2) wavelet propagating through a 2D series of layers. There is also a fault-like discontinuity near the middle of the field.

Needless to say, this is very early work. We are encouraged enough to be carrying on with new modeling code, but numerous challenges remain. Among them, more efficient code for parallel architectures, dealing with large datasets and persistent objects, increased breadth of application (we have largely restricted ourselves to finite difference code thus far), and acquisition of the necessary domain and mathematics knowledge.



Here we see the code distribution for the system thus far.

The domain knowledge is the fastest growing piece of the system. This includes mathematical ideas like how to turn a derivative into a finite difference, and physics ideas like wave propagation equations.

If you are trying to compare with the systems we have seen thus far, you will have noted there is no user interface portion. That is because we are borrowing it from Mathematica and other support systems.

# Reflections

## AI Techniques & Conventional Software

Domain Analysis, Reusability, Program Synthesis

## Representation, Inference, Interaction

## AI ≠ Expert Systems, Neural Networks

## AAAI: Re-engaging Theory & Application

NCAI ... IAAI

 Schlumberger Laboratory for Computer Science

I hope by now it is clear that considerable software action has taken place since the first electric log was hand-plotted in 1927. Keynote talks should leave the audience with some messages. So...I have four.

First, it is apparent to many who have attempted to apply AI techniques that this software must gracefully co-exist with the conventional software that after all solves most of the problems! (This is especially clear to AI businessmen.)

A trend to be encouraged is the exchange and sharing between software scientists who are concerned with knowledge engineering and those who are concerned with domain analysis, and between those who are concerned with reusable knowledge bases and those who are concerned with reusability in the large.

Continuing along this vein, it is difficult to determine the boundary between a "smart" compiler or an application generator and an automatic programming system. In any case, the goal is to continue to raise the level of abstraction at which we can interact with computers...it is always fun to remember that in the late fifties, Fortran was called an automatic programming system.

Second, I encourage you to include Interaction in your discussions about AI architectures—not as an afterthought, a minor "user interface" problem to be dealt with later—but as a first class citizen with Representation and Inference.

There is a good opportunity here for valuable research work. The AI literature is by no means saturated with high quality papers on this topic...and from an applications standpoint I hope the slides on where much of the code and design effort goes have convinced you that Interaction must be taken seriously.

My third point is simple—AI is something more than Expert Systems and Neural Networks. I think it is especially important to remember that the technology underlying commercial, not research, but commercial, expert systems is really quite simple. Much remains to be done...and we are starting to hear of interesting applications based on other AI technology. Two weeks ago, I attended the second conference on Innovative Applications of AI. This conference is sponsored by the national AI organization, the AAAI. The ground rules are that only fielded applications can be presented.

Over the last two years, we have heard a number of expert system presentations. However, we have also learned of systems based on truth-maintenance, case-based reasoning, and constraint-based reasoning.

Finally, I believe that the kind of applications work we see at this conference is essential to drive the AI field forward. It is largely through people like you putting the ideas of the theoreticians to the ultimate test of use in practice that new theory and techniques are developed.

I believe we are on the verge of a re-engagement of theory and application in the AAAI.

To this end, we will next year re-engage the National conference on AI—where we hope to see strong research contributions—with the Innovative Applications conference—where we hope to see fielded systems. The idea is that it is useful for AI scientists to hear first-hand the problems that appliers of the technology actually have.

A case in point is striking a balance between the time available to solve a certain problem vs the complexity of the reasoning required to achieve a certain level of performance. In fielded applications we begin to see that the constraint of time sometimes drives the invention of novel mechanisms for reducing the complexity of the reasoning...while maintaining the desired performance level.

(Reuters uses es to categorize news stories for Country Reports)

Thank you very much for your attention. I wish you a very successful conference. I would be happy to talk with any of you with questions or comments either now or later in the day.